

Introduction

The objective of this project was to determine the most proficient AI for playing Dubito. Within this endeavor, you'll discover both the game itself and the framework for conducting numerous experiments involving various AI players.

The Game

Dubito is a dynamic card game designed for 3-8 players. To kick off the game, cards are distributed in a round-robin fashion, starting with the initiating player.

Each turn, players have the option to either make a play or doubt the previous player's move. If there are no cards on the table, players are limited to making a play. When making a play, a player can choose to place 1-3 cards face down and declare a number (ranging from 1 to King). A truthful play occurs when the declared number matches the cards placed (e.g., declaring "1" and placing two cards of value 1), while any other declaration constitutes a bluff.

Alternatively, players can opt to doubt the previous player's claim. If the previous player was indeed bluffing, they collect all cards on the table, and the doubting player proceeds with their turn. Conversely, if the previous player was truthful, the doubting player collects all cards on the table, and the game continues with the next player.


The game culminates when only two players remain, resulting in the elimination of those two players and victory for the remaining participants.

Simplifications

To reduce the complexity of the experiments, certain simplifications were implemented:

- **Number of winner:** Limited to a single winner, as scenarios with multiple winners in a single game are treated as recursive instances of the "One winner case".
- **No jollies:** In order to reduce AI complexity.

AI

The AI comprises a straightforward tree structure that delineates various turn scenarios in Dubito based on provided information.  Dubito Tree AI

Input

To enable well-informed decision-making, a dictionary is conveyed to the AI at each turn. This dictionary contains the following information:

- **hand:** the cards of the current player

- **board_cards**: number of cards in the board (0 means you're the first)
- **playing_cards**: all numbers without discarded cards
- **current_number**: the card number called from the previous player (0 means you're the first)
- **n_cards_played**: number of cards played by the previous player
- **streak** : Number of turns without doubts
- **prev**: information about previous player
 - **n_cards**: amount of cards of this player
 - **turns**: how many turns the player played
 - **not_first_turns**: how many turns the player played (not first hand)
 - **doubts**: number of times the player doubted
 - **honest_times**: number of times recorded that this player was honest when doubted
 - **dishonest_times**: number of times recorded that this player was dishonest when doubted
- **next**: information about next player
 - **n_cards**: amount of cards of this player
 - **turns**: how many turns the player played
 - **not_first_turns**: how many turns the player played (not first hand)
 - **doubts**: number of times the player doubted
 - **honest_times**: number of times recorded that this player was honest when doubted
 - **dishonest_times**: number of times recorded that this player was dishonest when doubted

e.g. {'board_cards': 1, 'current_number': 1, 'n_cards_played': 1, 'streak' : 1, 'next': {'n_cards': 5, 'dishonest_times': 0, 'doubts': 8, 'honest_times': 0, 'id': 1, 'not_first_turns': 10, 'turns': 16}, 'playing_cards': [1, 7, 10, 11, 13], 'prev': {'n_cards': 7, 'dishonest_times': 1, 'doubts': 0, 'honest_times': 1, 'id': 2, 'not_first_turns': 8, 'turns': 12}}

Output

Instead, the output is a clearly defined dictionary containing three keys:

- **doubt**: boolean, can be either true or false
- **number**: represent the number that the player want to pick (only when he is first hand)
- **cards**: represent the cards the player want to play

e.g. { 'doubt': False, 'number': random.choice(input_player['playing_cards']), 'cards': picked_cards }